Pivotal

Cloud Native Architecture

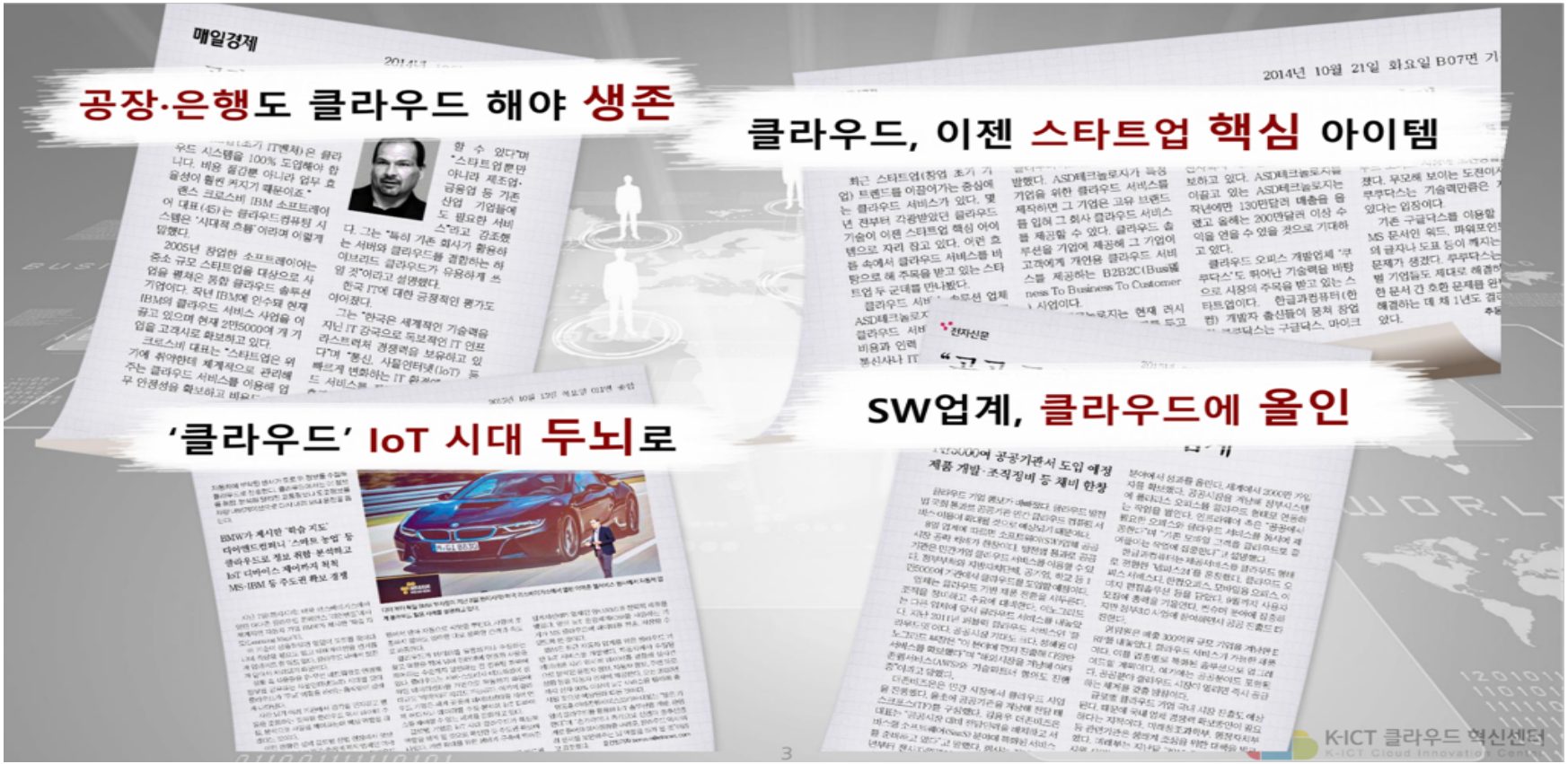
2018.1

Pivotal

클라우드 도입 현실화

Pivotal

언론에 비친 클라우드

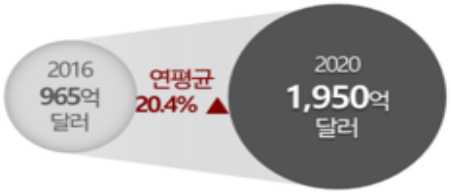


** 출처: 지능-정보-사회와-클라우드_클라우드혁신센터*

이슈를 넘어 현실화

전 세계 IT산업은 클라우드컴퓨팅 방식으로 IT 패러다임 변화

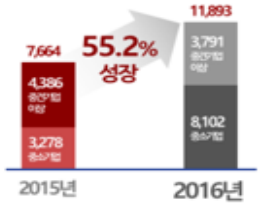
전 세계 퍼블릭 클라우드 서비스 시장 규모
2020까지 약 2배로 급성장
IDC 2016 년 예측



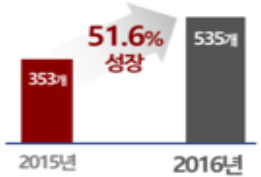
• 국내 클라우드 시장은 '16년 1.19조원이며, '18년은 1.97조원 예상

2016년 국내 클라우드 시장규모
약 1.19조원 | 전년대비 55% 성장
실태조사, NIPA

클라우드 시장 규모



국내 클라우드 기업 수



글로벌기업, 제조업, SW에 걸쳐
다양한 분야에 클라우드 확산

해외 클라우드 동향

다양한 산업 적용

- 아마존, MS, 세일즈포스닷컴 등 기존 강자는 IoT/빅데이터를 결합하여 제조업 등에 적용 중

※ 클라우드에 IoT를 적용할 수 있는 플랫폼으로 확대

산업 급성장

- 정부의 산업 육성 의지, 바이두/알리바바/텐센트 등 적극적 투자로 연평균 50% 성장 중

※ '15년 시장규모 : 16.3조원(중국 청과연구센터)

CSB 활성화

- 글로벌 기업과 자국의 SI 및 클라우드 기업이 공존하는 CSB 시장이 활성화 중

※ 시장규모(IDC재팬, '15년) : '14년 3.5조 → '19년 10.1조

산업의 기반기술

- 글로벌 제조사인 GE는 산업용 클라우드 개발*
- 구글(알파고), IBM(왓슨)은 클라우드로 A.I 구현

Pivotal

** 출처: 지능-정보-사회와-클라우드_클라우드혁신센터*

클라우드 주요 현상

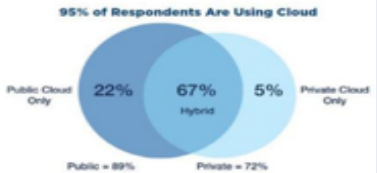
웹서비스 중심에서 모든 산업으로, 신규 구축에서 레거시 이동으로 클라우드 도입

클라우드 주요 현상

과거

웹서비스 · 게임 · 미디어	기존업체보다 스타트업	레거시보다 신규구축	국내서비스보다 글로벌서비스
			
Elastic 인프라 환경 Time to Market실현	클라우드 장점의 극대화	전환비용 없고, 리스크가 없음	글로벌 서비스는 레거시 방식의 가장 큰 혜택

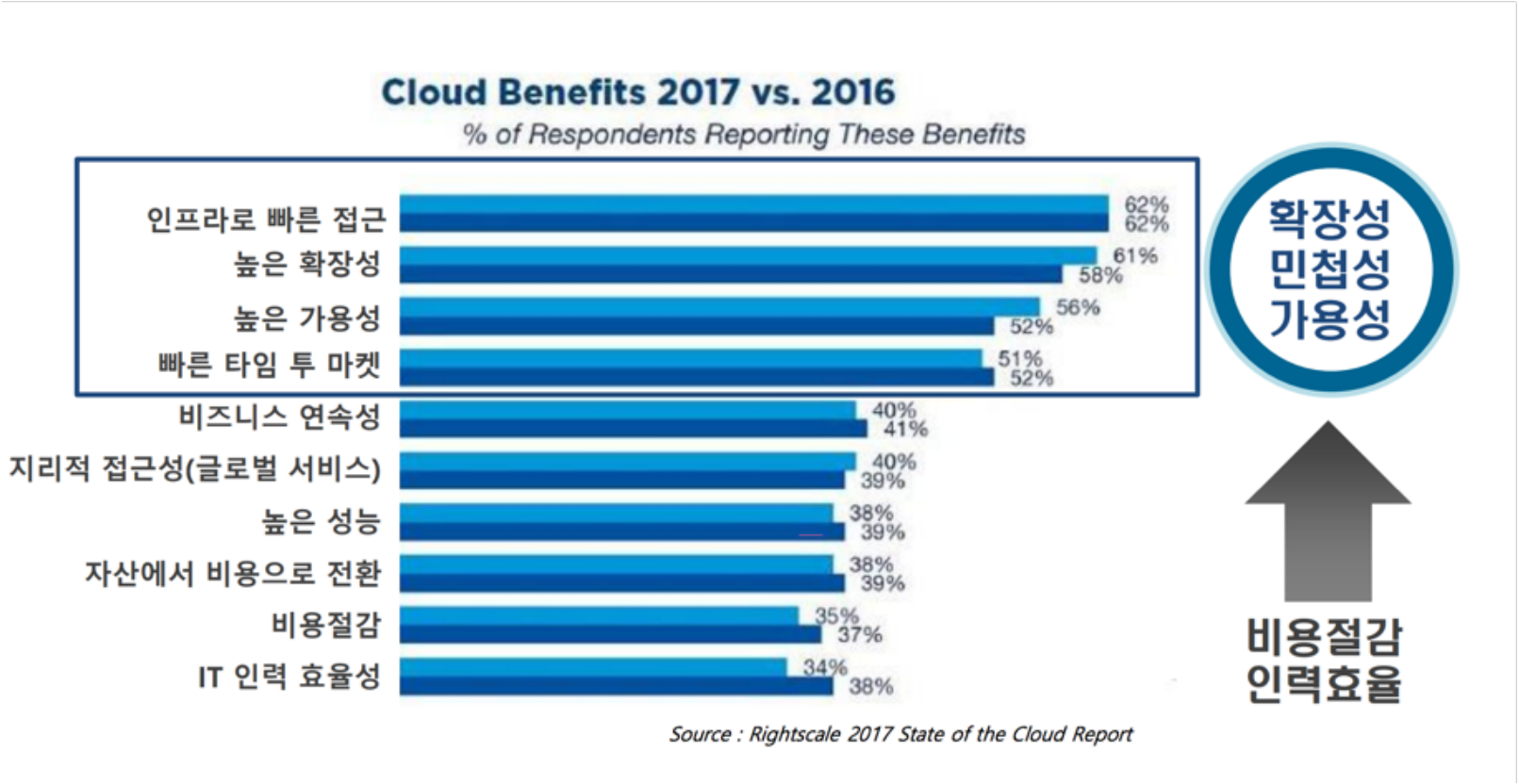
현재

법규제개선	공공 · 의료 · 금융분야 도입	보안을위해도입	하이브리드 구축
			
금융, 의료, 교육 등의 법규제개선	공공기관 클라우드 도입 활 성화	보다 강화된 보안과 시스템 안 정성을 위해 클라우드 전환	기존 레가시가 클라우드로 전환

Pivotal

** 출처: 지능-정보-사회와-클라우드_클라우드혁신센터*

클라우드 사용 이유



Pivotal

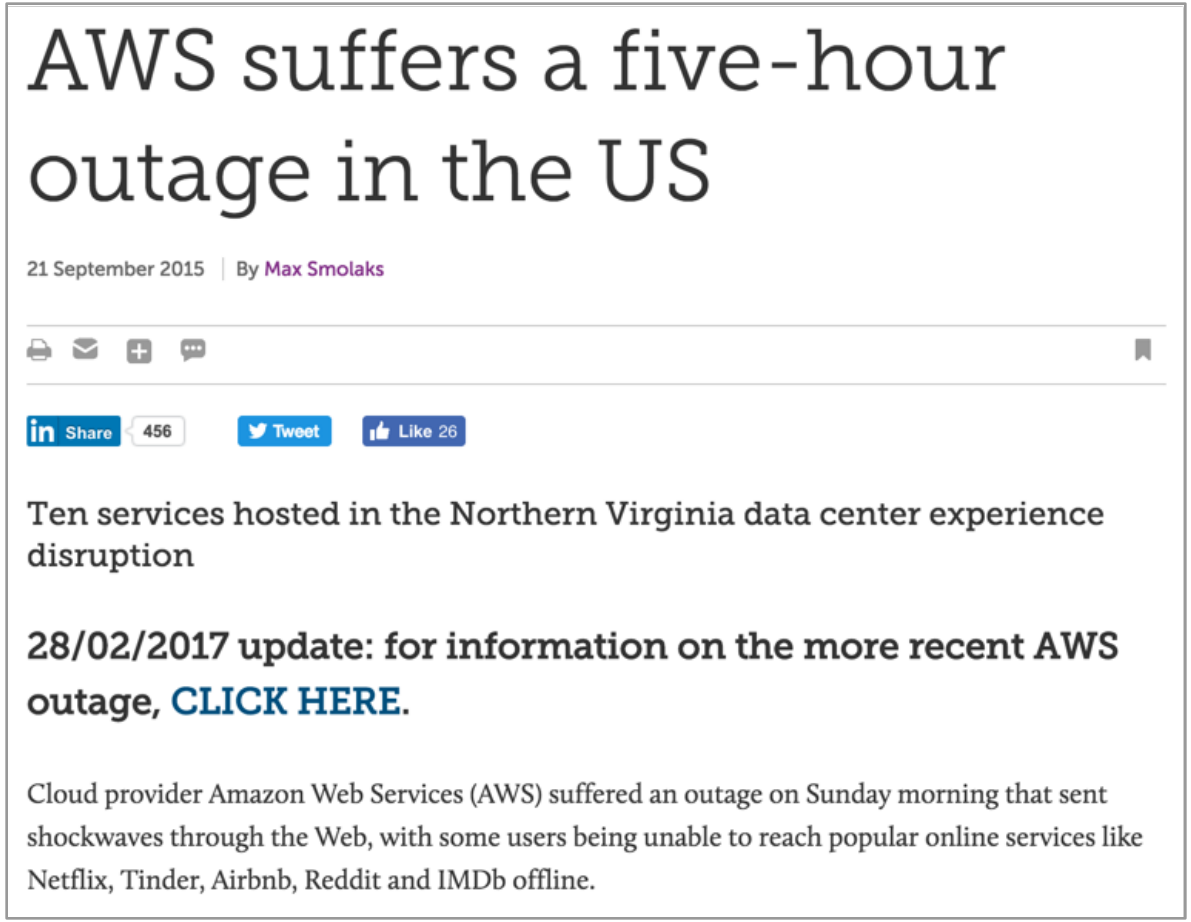
** 출처: 지능-정보-사회와-클라우드_클라우드혁신센터*

Why Cloud Native?

Scale and Resilience

모든 것은 항상 실패한다!

It's not Amazon's Fault



AWS suffers a five-hour outage in the US

21 September 2015 | By Max Smolaks

Share 456 Tweet Like 26

Ten services hosted in the Northern Virginia data center experience disruption

28/02/2017 update: for information on the more recent AWS outage, CLICK HERE.

Cloud provider Amazon Web Services (AWS) suffered an outage on Sunday morning that sent shockwaves through the Web, with some users being unable to reach popular online services like Netflix, Tinder, Airbnb, Reddit and IMDb offline.

탄력적인 확장

자동/수동 수평 확장/축소

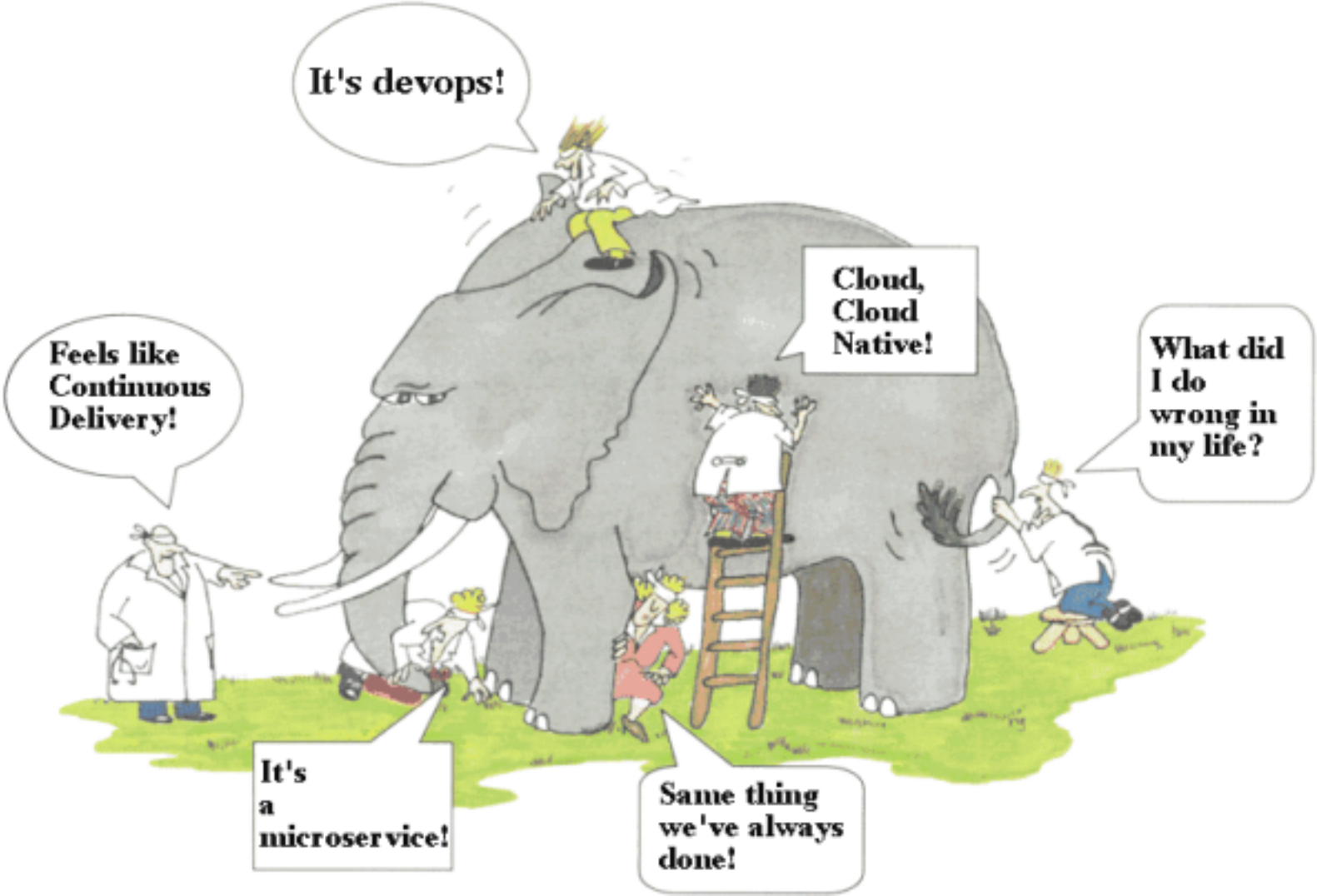
- 구성 변경이 잦음

- **실행 앱 인스턴스(VM/Container) 개수 탄력적 변경**
  - LB에 추가 인스턴스 등록/삭제?
  - API 호출 시 서비스 서버 목록 관리는?
  - 모니터링은?

- **시스템의 구성 정보(IP, Hostname, ...) 동적 변경**
  - 어떻게 서비스를 찾을 것인가?
  - 기존 모니터링 시스템에서 관리가 가능한가?

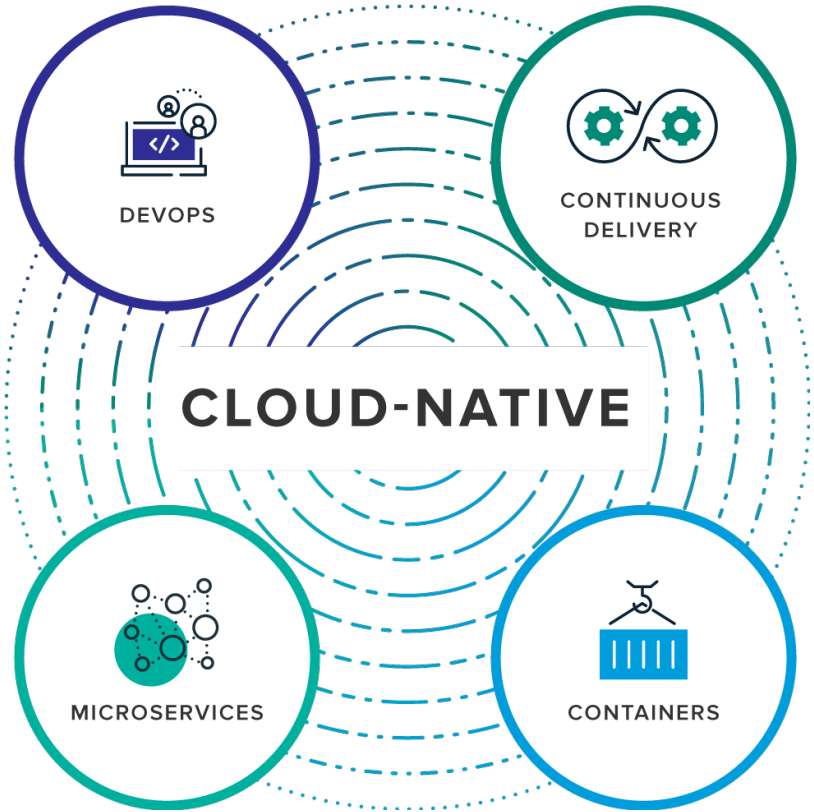
- **Multi-Cloud 고려**
  - Hybrid Cloud 또는 Multi Vendor Cloud 구조 확장시?

What is Cloud Native Architecture?


It's devops!
Cloud, Cloud Native!
Feels like Continuous Delivery!
What did I do wrong in my life?
It's a microservice!
Same thing we've always done!

클라우드 컴퓨팅 모델의 장점을 모두 활용하는 애플리케이션을 개발하고 실행하기 위한 접근 방식

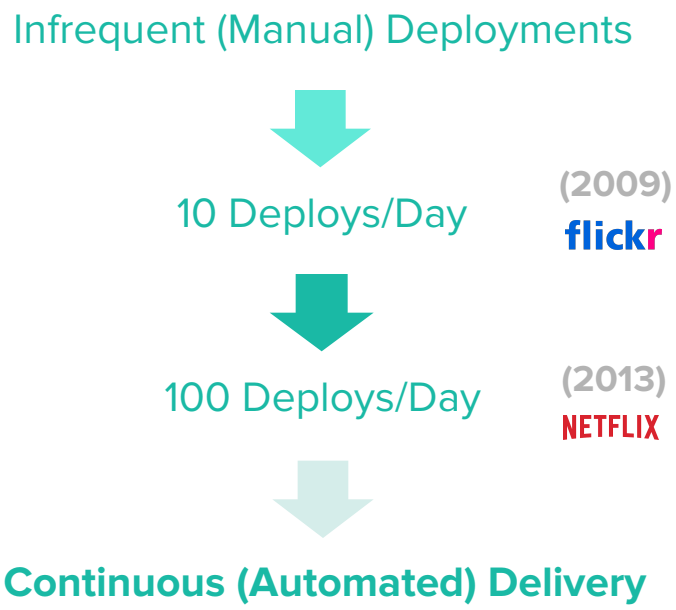
Cloud Native 핵심 요소



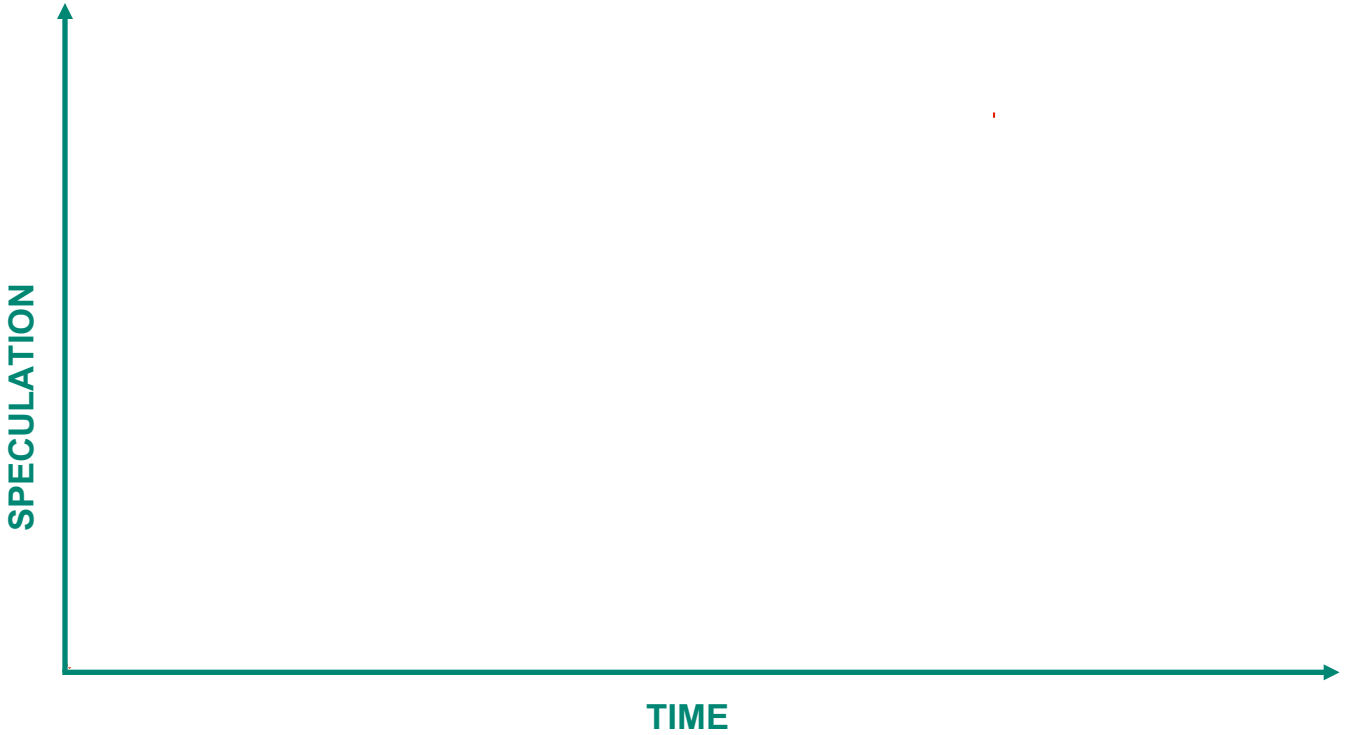
지속적인 서비스 딜리버리
(Continuous Delivery)

A Seattle book store deploys code, on average, every second

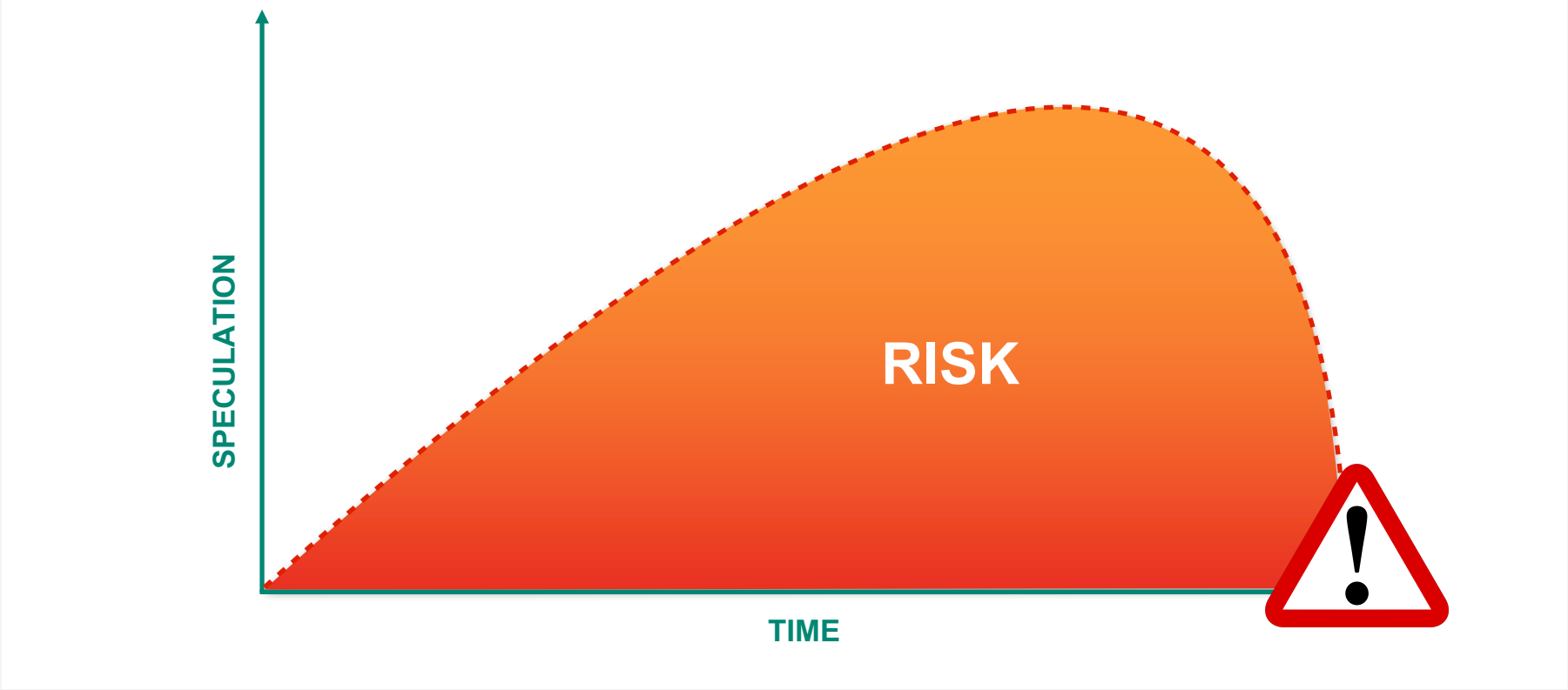
Software Cycles Have Been Getting Faster



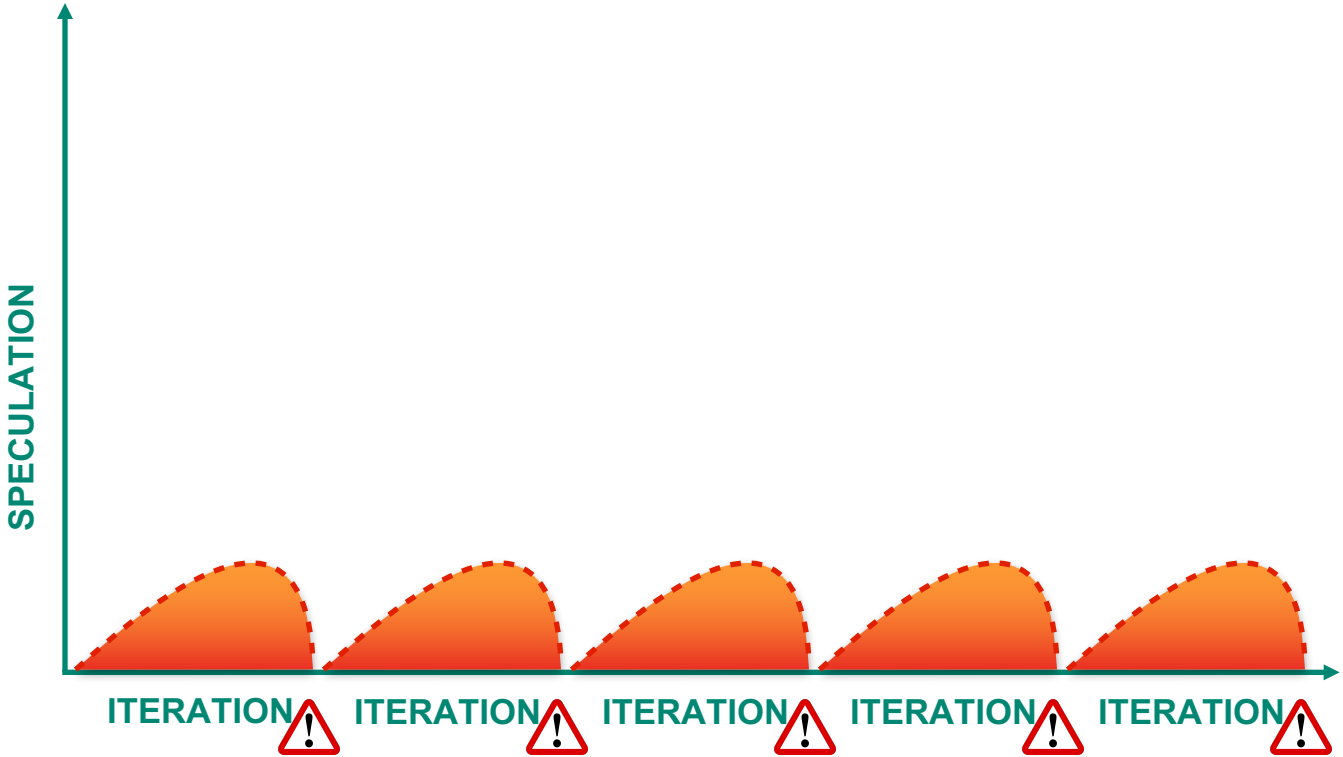
Why?



Risk Accumulation



Experimentation Reduces Risk



Continuous Delivery

of

Customer Value

Obstacles

- Silos: Dev, QA, Operations is typical. No shared common goal
- Dissimilar Environments - “It works on my machine”
- Risky Deployments: Manual steps, done “off hours”
- Changes are treated as an exception, not the norm
  →Firefighting
- Processes designed around these obstacles

7 Prerequisites for Continuous Delivery

1. DevOps culture
2. Continuous Integration
3. Automate and version tasks, for auditing & reproduction
4. Shared tools & procedures
5. Apps must be easy to deploy to prod & easy to rollback
6. App must be deployed to programmable target
7. Application versions must be ready to be shipped into production

Continuous Integration & Delivery

Deliver High Quality Software Faster & Continuously, from Idea to Production

AUTOMATION.
Integrate tools and automate processes from testing to builds and deployment

SPEED.
Release more frequently with smaller bits will reduce complexity and improve time-to-market

QUALITY.
Reduce feedback loop using test-driven development to surface problems sooner and be responsive

AGILITY.
Push updates on regular basis with no downtime to improve customer experience and time to market

Build Pipeline Operations

Commit Code Change → **Automate Build & Test** (Unit Test, Static Code Analysis) → **Store Binaries & Build Artifacts** → **Automated Integration Testing** → **Acceptance, Performance & Load** → **Zero Downtime Upgrade to Production**

Tool Chain

GitlLab

Distributed revision control and source code management. Collaborative software development

Concourse

Build and test software projects continuously and incrementally. Hundreds of compatible plugins

jFrog

Share binaries and manage distributions. Manage artifact lifecycle. Avoid license violations

Development

Test + UAT + Staging

Production

Develop, test, QA and production on the same platform. Simple, developer friendly commands and APIs. Operational benefits for every app. Built-in ecosystem services. Deploy, operate and scale on any IAAS

데브 옵스
(DevOps)

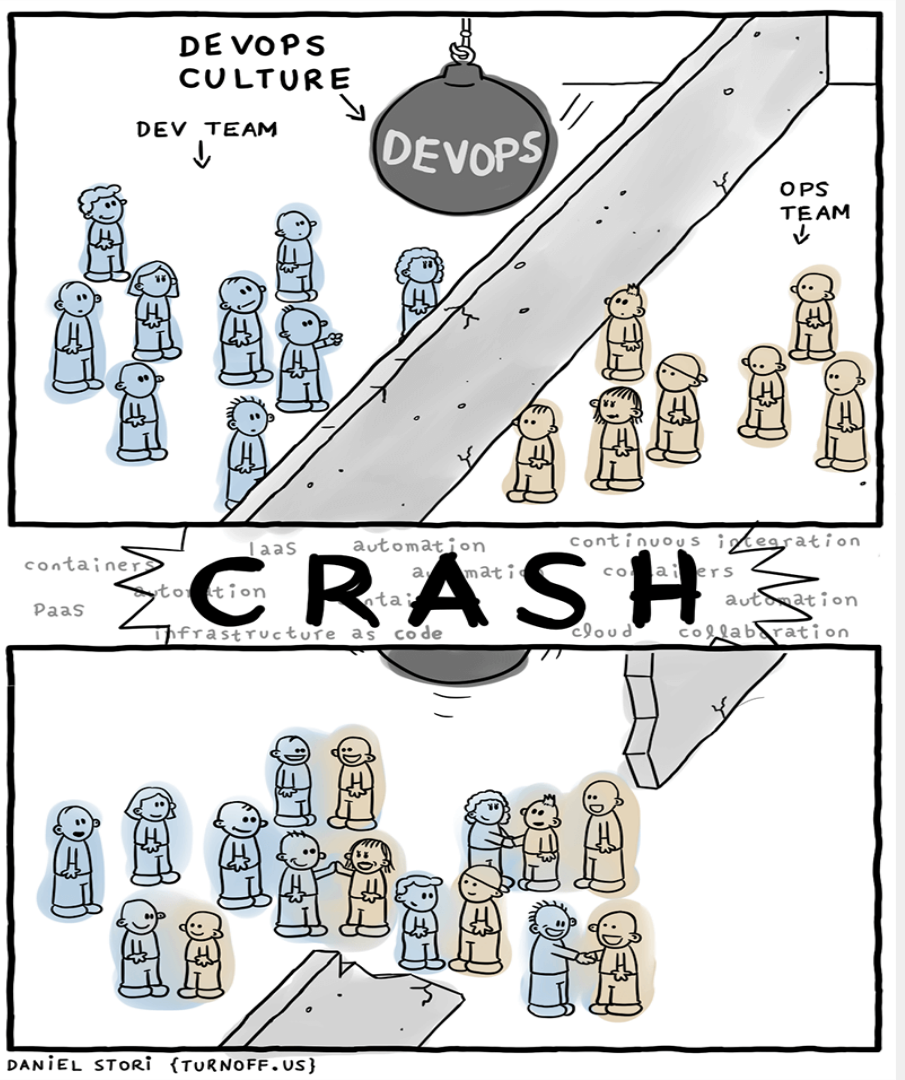
The History of IT Organizational Systems

- 1960-1980s: ***Fragile*** ***"Cowboys"***
  - Optimize around the **hardware**
- 1990-2000s: ***Robust*** ***"ITIL"***
  - Optimize and protect around **changes** or **traumas**
- 2010s+: ***Resilient/Anti-Fragile*** ***"DevOps"***
  - Optimize on continuously improving the **flow of value**

DevOps?

- **DevOps** is **a way of working** that focuses on regularly **shipping quality software that meets a business need ...**

- **DevOps** is fluid, and **avoids** much of the **traditional handoff friction and delays** between product development and IT operations through ...



Pivotal

피자 두판의 팀이란?

각 팀이 분산된
민첩하면서, 독립적인,
신뢰하고, 오너쉽을
가진 서비스 팀

"DevOps"

Example Platform Team: Pivotal CloudOps

~6000+ Application Servers: 7 Full Time Staff and 1 Rotation

Pivotal

마이크로 서비스
(Micro Services)

July 15th is Prime Day Learn More

amazon Try Prime

All 11 seconds

Search μservice

Shop Beautiful Things, Updated Daily

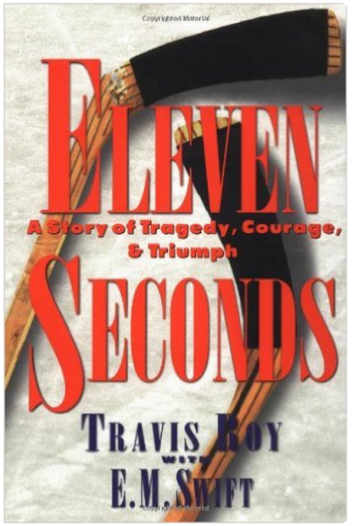
Shop by Department | Your Amazon.com | Today's Deals | Gift Cards | Sell | Help

Hello. Sign in Your Account | Try Prime | Wish List | 0 Cart

Shopping Cart μservice

Books | Advanced Search | New Releases | Best Sellers | The New York Times® Best Sellers | Children's Books | Textbooks | Textbook Rentals | Sell Us Your Books | Best Books of the Month | Deals in Books

Look inside



Flip to back

See all 3 images

Image μservice

Eleven Seconds: A Story of Tragedy, Courage & Triumph

Hardcover – January 1, 1998

by Travis Roy (Author), E. M. Swift (Author)

Item Master μservice

4.5 stars 62 customer reviews

Reviews μservice

#1 Best Seller in Hockey Biographies

See all 4 formats and editions

Kindle	Hardcover
$8.05	$16.85
Read with Our Free App	118 Used from $0.01 48 New from $3.95 14 Collectible from $4.95

In this heartfelt testament to the power of love and the strength of the human spirit, Travis Roy, who suffered a devastating injury eleven seconds into his first college hockey game, reveals how he has managed to cope after the accident and, with the help of family and friends, overcom tremendous barriers to begin a new life.

Best of the Month

See the Best Books of the Month

Want to know our Editors' picks for the best books of the month? Browse Best Books Month, featuring our favorite new books in more than a dozen categories.

Share

Buy New $16.85

Qty: 1

List Price: $26.00

Save: $9.15 (35%)

FREE Shipping on orders over $35.

Only 18 left in stock (more on the way).

Ships from and sold by Amazon.com.

Gift-wrap available.

Yes, I want **FREE Two-Day Shipping** with Amazon Prime

Add to Cart

Turn on 1-Click ordering for this browser

July 10? Order within 3 hoose Two-Day ut. Details

ddress:

Other dependent μservice

Microservices

If services have to be updated together, they're not loosely coupled!

Loosely coupled service oriented architecture with bounded contexts

- Adrian Cockcroft

If you have to know about surrounding services, you don't have a bounded context

"Bounded Context는 독립적으로 서비스 될 때 문제없는 업무 범위"

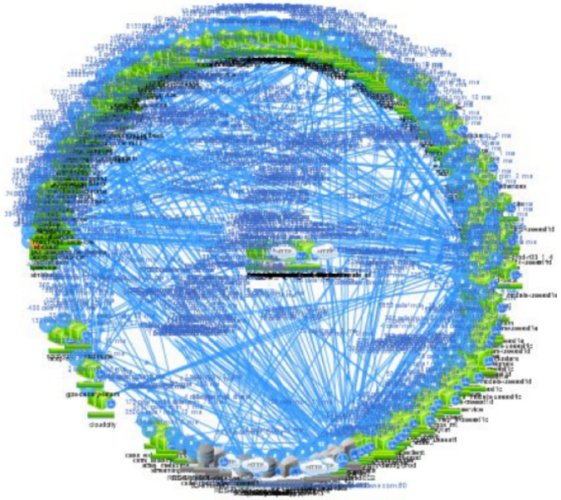
Pivotal

Microservices

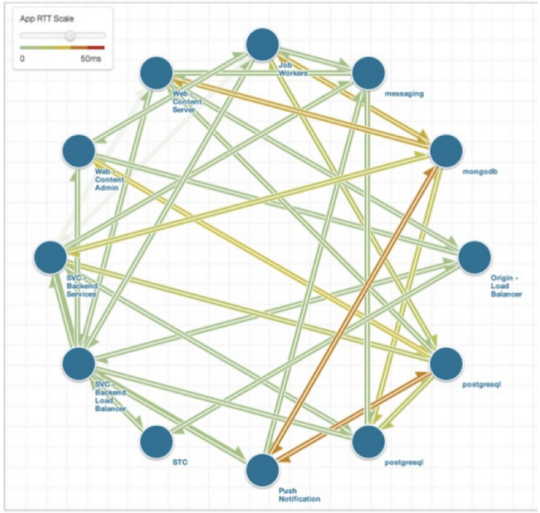
"각각의 기능별로 독립적인 서비스로 구성하여 작은 서비스의 집합으로 원하는 기능을 만들어 내는것"

Pivotal

Microservice Pioneers



Netflix



Gilt Groupe (12 of 450)

Twitter

Pivotal

Who Moved My Complexity?

- Microservices are individually simple
- Complexity is transferred to the ecosystem

Pivotal

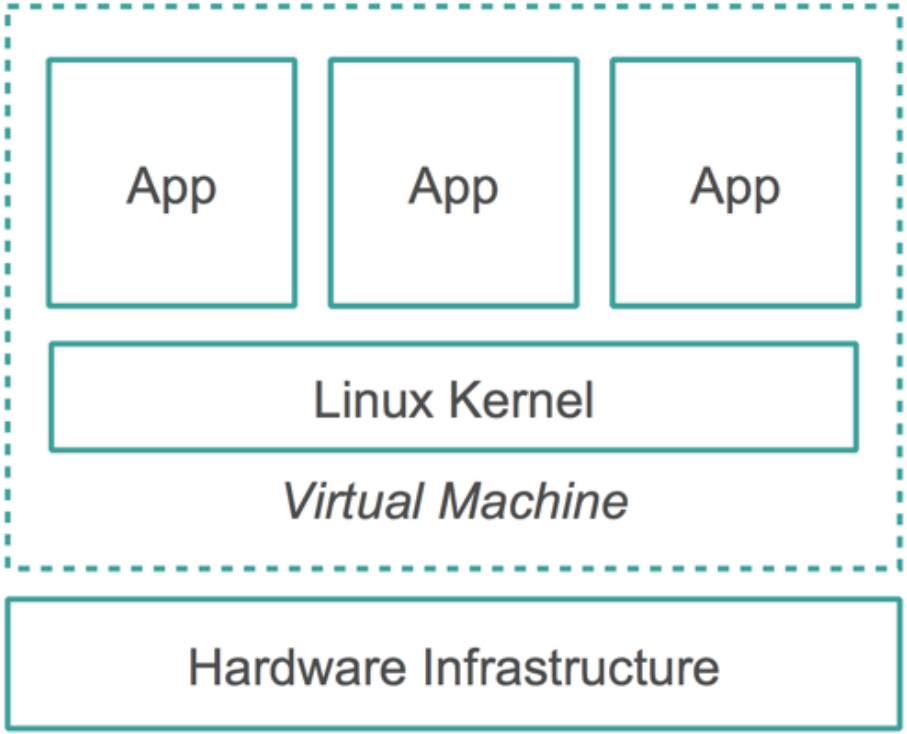
Challenges in a Distributed System

- Configuration management
- Registration and discovery
- Routing and load balancing
- Fault tolerance and isolation
- Aggregation and transformation
- Monitoring and distributed tracing
- Process management

Pivotal

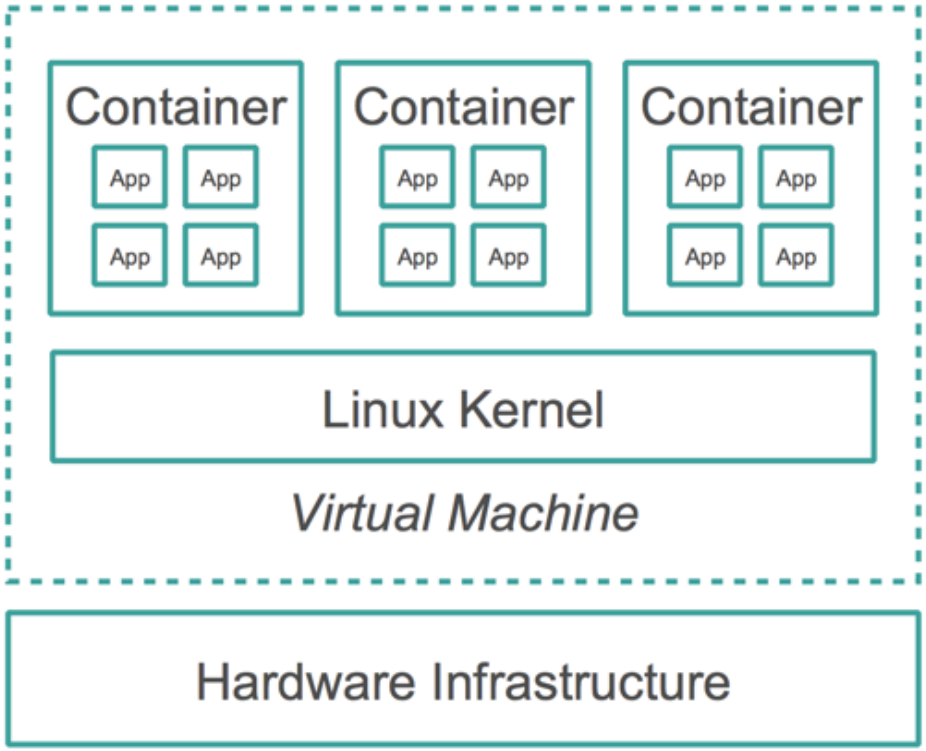
컨테이너
(Container)

Application Server Deployment - Monolith



- Load balancing requires provisioning of new VMs and app server installations
- Poor resource isolation; memory leaks can cause other applications to become unavailable
- Runtime environment is driven by the operator

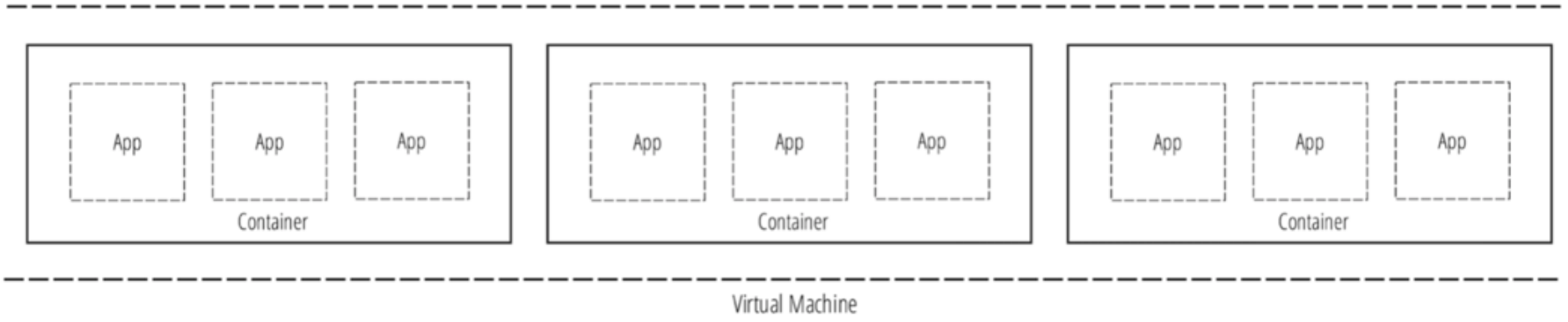
Linux Container Deployment - Microservice



- Development team drives the application runtime of a container
- Containers are resource isolated, allowing efficient scheduling onto a grid of VMs
- Containers take seconds to start, VMs take minutes

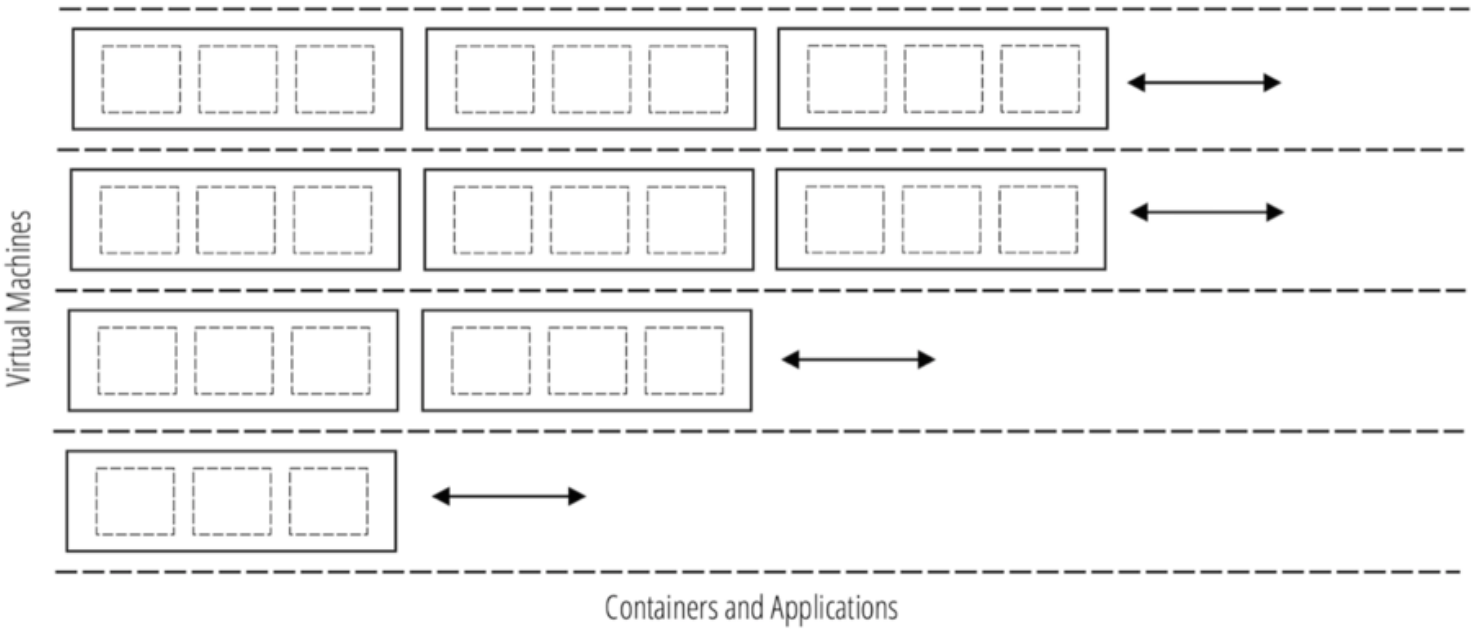
Microservices - Container Deployment

- Each microservice can be containerized with their application dependencies
- Containers get scheduled on virtual machines with an allotted resource policy



Container scheduling and auto-scaling

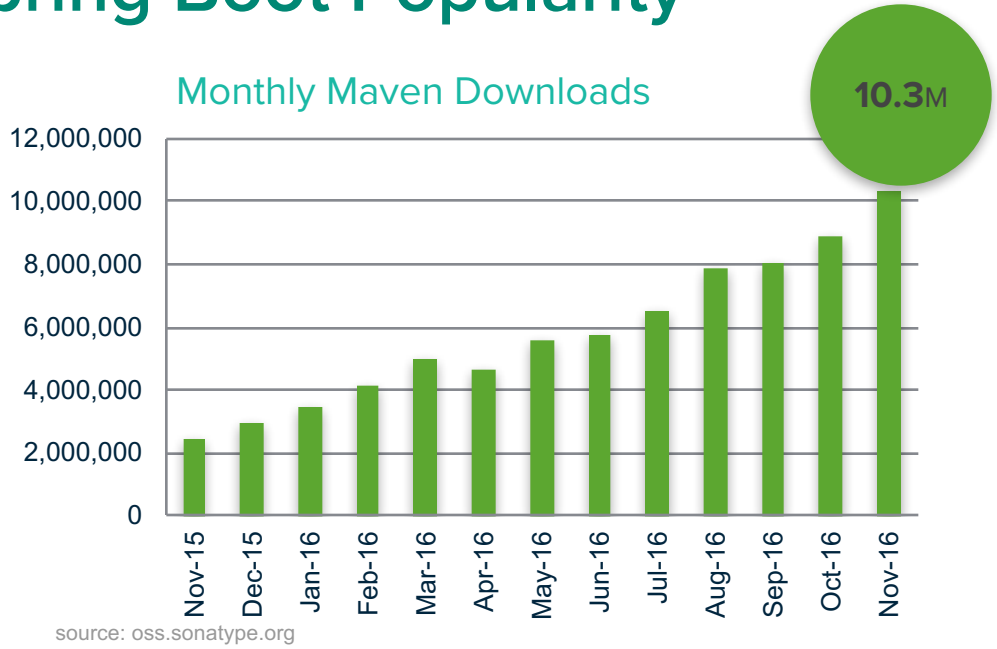
- Minutes to start a VM, but seconds to start a container
- An elastic runtime can perform auto-scaling of containers in PaaS layer



How?

- Tools
- Platform
- Process and Culture

Spring Boot Popularity



source: oss.sonatype.org

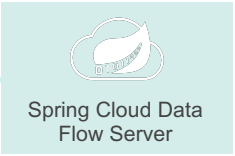
Spring Cloud – Microservices Infrastructure

Common Patterns for Distributed, Cloud Based Enterprise Applications

Designed for disposable infrastructure in partnership with Netflix

Spring Cloud Data Flow – Data Microservices

Deploy and Management on PCF



Orchestration Service for Data Microservices

Developer Benefits

- ✓ Simple programming model
- ✓ Composition: Streaming and Batch
- ✓ Readily Available Building Blocks
- ✓ Messaging Middleware
- ✓ CI/CD + TDD Friendly

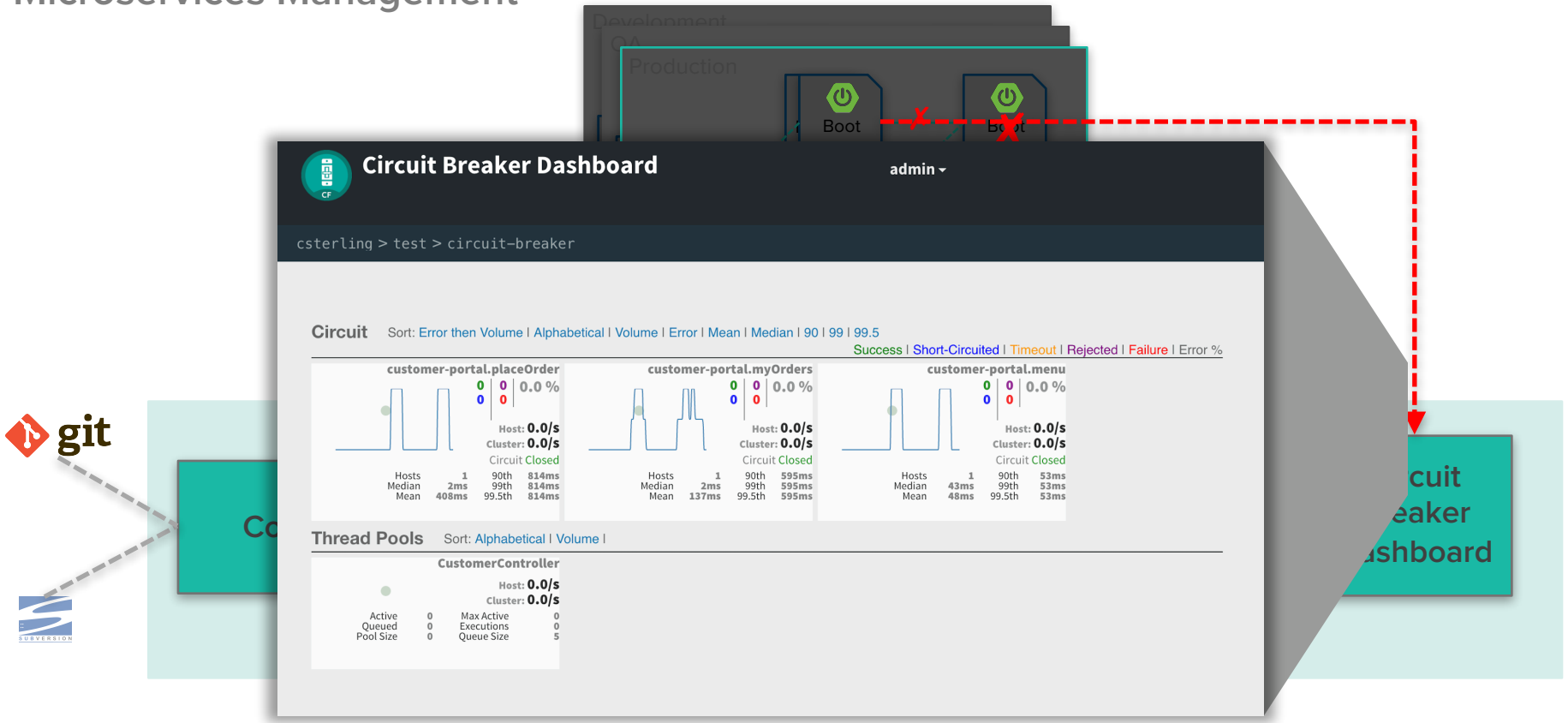
Operator Benefits

- ✓ Canary/rolling upgrades
- ✓ Dynamic failover and auto scaling
- ✓ Easy Failure Isolation
- ✓ Operational Dashboard

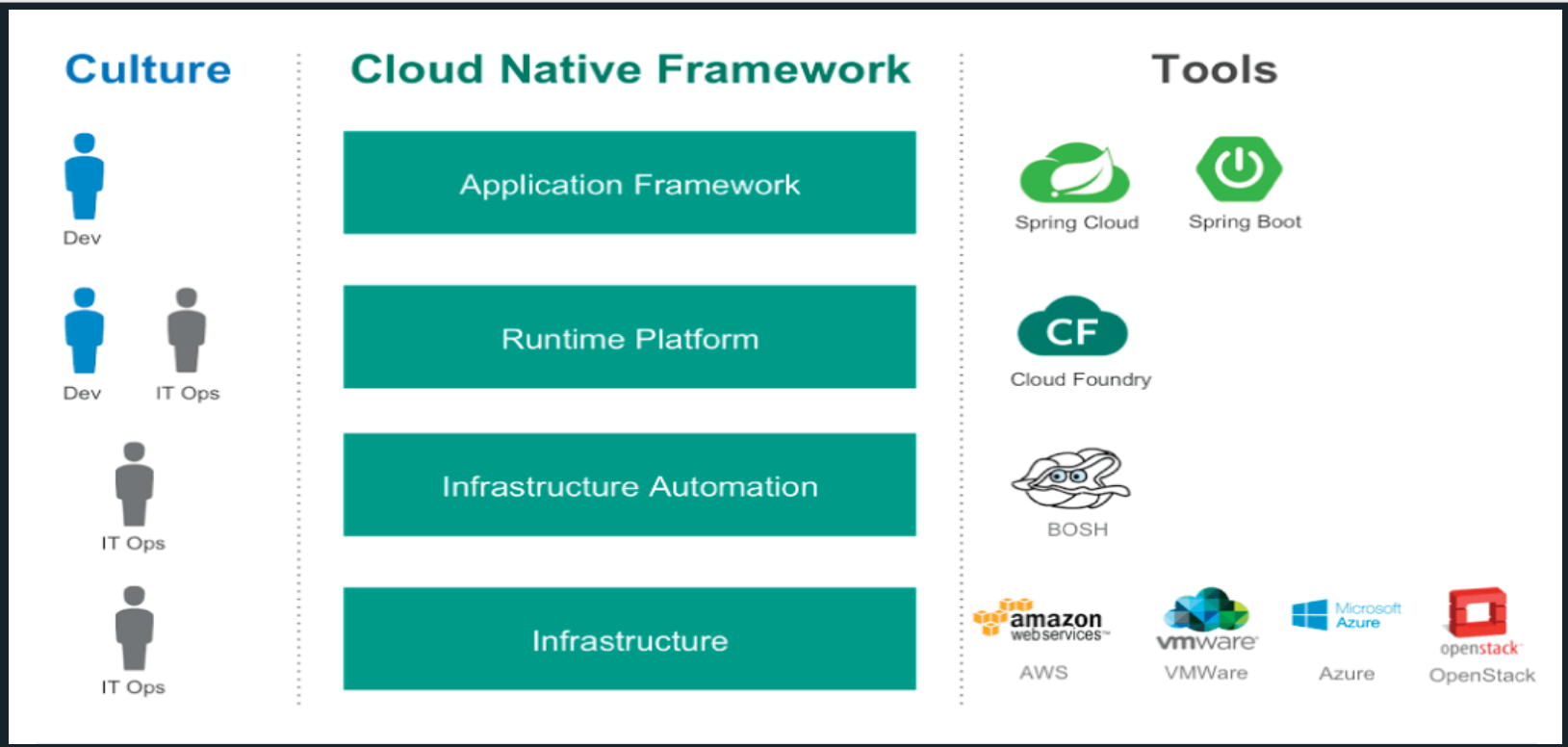
HTTP

Spring Cloud Services

Microservices Management



Pivotal Cloud Foundry



Pivotal

The Journey: Technology

- Provide the right tools
  - Platform, design pattern foundations, continuous delivery…
- Start small
  - Right-size for your needs
  - For migrations, don’t boil the ocean
- Target applications that:
  - Have high business/competitive impact
  - Need to scale or change frequently

Pivotal

The Journey: Culture & Organization

- Small, cross-functional teams
- Products, not projects
- Test-driven development
- Short iterations
- Transparency and visibility

Pivotal

Transforming How The World Builds Software